

## 20

01/02/2002

This Page Blank (uspto)

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平9-319530

(43) 公開日 平成9年(1997)12月12日

(51) Int.Cl.<sup>6</sup>

識別記号

庁内整理番号

F I

技術表示箇所

G 0 6 F 3/12

G 0 6 F 3/12

C

B 4 1 J 29/38

B 4 1 J 29/38

T

Z

審査請求 未請求 請求項の数2 F D (全 16 頁)

(21) 出願番号 特願平8-152912

(22) 出願日 平成8年(1996)5月27日

(71) 出願人 000005496

富士ゼロックス株式会社

東京都港区赤坂二丁目17番22号

(72) 発明者 藤本 厚史

神奈川県川崎市高津区坂戸3丁目2番1号

K S P R & D ビジネスパークビル 富

士ゼロックス株式会社内

(74) 代理人 弁理士 南野 貞男 (外1名)

(54) 【発明の名称】 プリントドライバ

(57) 【要約】

【課題】 複数のページ記述言語に対応するプリンタを利用する場合に、プリントドライバの側でデータ転送効率やデータ処理量および画像品質の点から最も適切なPDLを自動で選択することのできるプリントドライバを提供する。

【解決手段】 プリンタを制御するプリントドライバは、印刷データの特徴を判断する判断手段と、制御するプリンタで使用できる複数の種類のページ記述言語のうち最適なページ記述言語の種類を前記判断手段の判断結果に基づき選択する選択手段と、選択された種類のページ記述言語で前記印刷データを表現した出力データを作成する作成手段とを備える。選択手段は、過去の出力データの作成時間、前記出力データのプリンタへの転送時間を参照して、最適なページ記述言語を選択する。

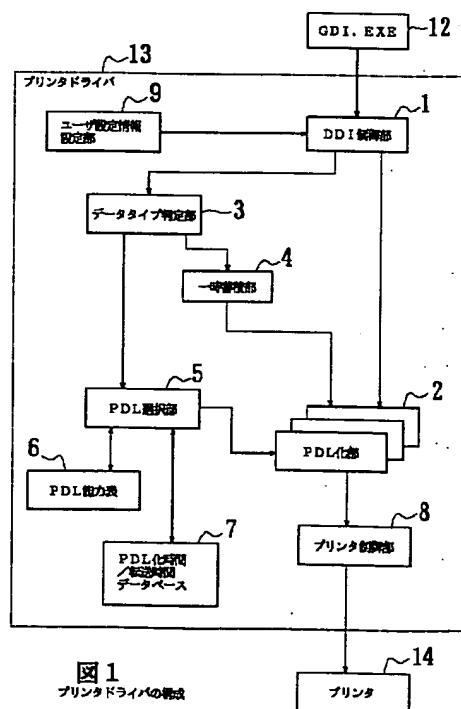


図1  
プリントドライバの構成

## 【特許請求の範囲】

【請求項1】 印刷データの特徴を判断する判断手段と、

制御するプリンタで使用できる複数の種類のページ記述言語のうち最適なページ記述言語の種類を前記判断手段の判断結果に基づき選択する選択手段と、

選択された種類のページ記述言語で前記印刷データを表現した出力データを作成する作成手段とを備えることを特徴とするプリンタドライバ。

【請求項2】 請求項1に記載のプリンタドライバにおいて、

前記選択手段は、過去の出力データの作成時間、前記出力データのプリンタへの転送時間を参照して、最適なページ記述言語を選択することを特徴とするプリンタドライバ。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は、ページ記述言語を使用してプリントイメージを作成し印刷するプリンタのためのプリンタドライバに関し、特に、複数の種類のページ記述言語を解釈できるプリンタのためのプリンタドライバに関する。

## 【0002】

【従来の技術】昨今、プリンタの高機能化に伴って、複数のページ記述言語に対応できるプリンタが登場してきている。パーソナルコンピュータ上で動作するアプリケーションプログラムから、そのアプリケーションプログラムで作成した印刷データをプリンタに出力する場合を考えると、図6に示すように、プリンタ14が解釈するページ記述言語（以下、PDLと略称する）の描画命令は、パーソナルコンピュータ10とプリンタ14との間で受け渡しするための1種の画像情報の表現形式と見なすことができる。

【0003】PDLの描画命令を用いることにより、印刷する画像情報をそのままプリンタに転送する場合に比較して、データ転送量を減すことができる。また、標準化されたPDLを用いることにより印刷データの互換性を維持することができる。更に、プリンタが複数の種類のPDLの描画命令を受け入れる（解釈できる）ようにすることによって、パーソナルコンピュータ上のより広範なアプリケーションプログラムが生成する印刷データを扱えるようになる。

【0004】ところで、マイクロソフト社のオペレーティングシステム「WINDOWS」においては、図7に示すように、オペレーティングシステム「WINDOW S」上のアプリケーションプログラム11から送出される印刷データは、GDI(Graphics Device Interface)と呼ばれる共通のインタフェースのデータ形式により送出される。これをデバイスインタフェース処理部(GDI. EXE)12が受け取り、印刷データの画像情報を

プリンタドライバ13に渡す。この処理を行うデバイスインタフェース処理部(GDI. EXE)12は、「WINDOWS」オペレーティングシステムに含まれている共通のインタフェース処理機能である。つまり、デバイスインタフェース処理部(GDI. EXE)12が、アプリケーションプログラム11から送出される印刷データの画像情報を受け取り、プリンタドライバ13に適合するように形式変換して画像情報を渡す。プリンタドライバ13は、物理的に接続されているプリンタ14に対して、当該画像情報をそのプリンタに対応するPDLの描画命令や圧縮イメージに展開してプリンタ14に送る。プリンタドライバ13は、物理的に接続されるプリンタ14に適合するものを用意しておくことになる。

【0005】プリンタへのデータ転送時間を短縮するため、プリンタに印刷データを送る場合には、パーソナルコンピュータ内部で印刷データのデータ圧縮を行って一度蓄積してから送信する。このとき、例えば、特開平6-292024号公報に記載の「画像形成システム装置」のように、プリンタへのデータ転送時間を更に短縮するため、圧縮効率が悪いと判断される場合には、他の効率的な圧縮方法に当該印刷データのデータ圧縮方法を切り替えて圧縮する。このようにして、最適な圧縮方法を使用することによって、プリンタへの転送時間を短縮する。

【0006】この場合、プリンタに転送される画像情報の圧縮データを、パーソナルコンピュータとプリンタとの間で受け渡しするための1種の画像情報の表現形式と見なすと、圧縮方法を選択することは、画像情報の1つの表現形式を選択するという意味と同様である。画像情報の1つの表現形式を選択するという意味では、マルチPDL対応のプリンタにおいて、印刷データの表現形式の中の複数のPDLの形式から1つのPDLの形式を選択することと等価であると見なせる。

【0007】ところで、複数のPDLに対応できるマルチPDL対応プリンタの場合、PDLごとに別々のプリンタドライバが提供されるのが通例であるので、上記の技術を応用する場合、プリンタへのデータ転送時間を短縮するために、PDLによるデータ量の圧縮を評価(判断)して、適切なPDLを用いるプリンタドライバを選択することになる。

【0008】また、プリンタドライバを選択する技術に関しては、例えば、文献「“HP Bids Farewell to 300 dpi with New 4 ppm LaserJet 5L” THE HARD COPY OBSERVER, Oct. 1995」に記載されている内容が参照できる。この文献によると、プリンタに対応して、そのプリンタドライバをインストールする際には、プリンタドライバが動作する環境に応じて、例えば、オペレーティングシステムがマイクロソフト社の「DOS」なのか「WINDOWS」なのかに対応して、また、CPUの速度の大小、搭載しているメモリ量の大小などに応じて、イ

インストールするプリンタドライバを選択する技術が紹介されている。例えば、オペレーティングシステムが「WINDOWS」で、高速なCPUを使用し、多くのメモリを搭載してあるシステムの場合には、処理するデータ量のより多い「Windows Printing System」用のプリンタドライバをインストールし、オペレーティングシステムが「DOS」環境であれば、処理するデータ量の少ないPDL用のドライバをインストールする。

【0009】また、同文献によれば、アプリケーションプログラムの種類を認識し、そのアプリケーションプログラムによってプリンタドライバの動作パラメータを自動的に変更する技術についても紹介されている。これによると、例えば、マイクロソフト社の商品名「Microsoft Power Point」のようなグラフィックスを中心に扱うアプリケーションプログラムの場合には、グラフィックスの再現性を重視するパラメータを選択し、マイクロソフト社の商品名「Microsoft Word」のようにテキストを中心に扱うアプリケーションプログラムの場合には、テキストの再現性を重視するパラメータを選択する。

【0010】

【発明が解決しようとする課題】ところで、前述の特開平6-292024号公報に記載の「画像形成システム装置」による技術をプリンタドライバに適用する場合においては、プリンタ側に転送する画像データを実際に圧縮してみないと、どの圧縮方法が圧縮効率が高いかわからない。このため、何回も圧縮しなければならない場合があり、その変換に多くの時間がかかるという問題がある。

【0011】また、この特開平6-292024号公報に記載の「画像形成システム装置」の技術におけるデータ圧縮によるデータの表現形式の変換を、PDLの描画命令への変換(PDL化)に読み替えて、マルチPDL対応プリンタのプリンタドライバにおける最適なPDLの種類を選択することに適用すると、その場合、PDL自身の表現能力に差があるため、単純に圧縮効率だけによってPDLの種類を選択すると、画質が劣化する場合があるという問題が生じる。

【0012】前述したように、アプリケーションプログラム11(図7)から出力される印刷データによる画像情報(描画コマンド)は、デバイスインタフェース処理部(GDI.EXE)12が受け取り、プリンタドライバ13に適合するような形式に変換して渡すので、プリンタドライバ13が、接続されているプリンタ用のPDL命令や圧縮イメージに展開して、プリンタ14に送る処理を行う。その場合、アプリケーションプログラム11からは、デバイスインタフェース処理部(GDI.EXE)12を介して、プリンタドライバ13に描画コマンドを発行するので、プリンタドライバ13は、デバイスインタフェース処理部(GDI.EXE)12からの描画の指示を受け取り、その指示をPDL命令に対応づ

けることになる。

【0013】その場合、プリンタに対応したプリンタドライバ13において、使用するPDLの描画処理の機能が十分であれば、図8に示すように、動作開始時に、プリンタドライバ13では、デバイスインタフェース処理部(GDI.EXE)12から描画機能の問い合わせコマンド(Enable)があると、これに対して、その能力があることを宣言する(GDIINFO)ので、これにより、例えば、アプリケーションプログラム11からの楕円描画命令「Ellipse」がデバイスインタフェース処理部(GDI.EXE)12を介して、そのままの形式「OUTPUT(Ellipse)」プリンタドライバ13に伝え、プリンタドライバ13から楕円描画のためのPDL命令(楕円描画)が出力される。

【0014】しかし、プリンタドライバのPDLの描画処理の機能が十分でない場合には、図9に示すように、動作開始時に、プリンタドライバ13は、デバイスインタフェース処理部(GDI.EXE)12から描画機能の問い合わせコマンド(Enable)があっても、その能力がないことを宣言する(GDIINFO)ことになるので、例えば、アプリケーションプログラム11から送出された楕円描画命令は、デバイスインタフェース処理部(GDI.EXE)12によって複数のライン描画命令「OUTPUT(SCANLINE)」に置き換えられて描画指示される。この場合、プリンタドライバ13は、そのライン描画命令「OUTPUT(SCANLINE)」からのPDL命令への置き換えは、必ずしも接続されているプリンタに適切な形で分解されないため、もともとPDL命令によって楕円描画がサポートされている場合に比較して描画品質が大きく劣化する。

【0015】また、上記の文献に記載の技術のように、インストールする際にシステム的环境に応じて、プリンタドライバを選択する方式を用いると、インストール時には確かに動作環境に応じたPDLの選択は可能であるが、インストール後は常に同一のプリンタドライバが使用されるため、印刷データに応じた動的なPDLの選択は行えないという問題がある。

【0016】また、同じく、上記文献に記載の技術のうち、アプリケーションプログラムによって動作パラメータを変更する方式によると、1種類のPDLへの変換パラメータをアプリケーションプログラムに応じて変えているだけであるため、その場合、プリンタ側に複数のPDL生成部が実装されていても、それら複数のPDLを選択的に使用できず、印刷データに適切なPDLの選択までは行えないという問題がある。

【0017】ここでは、変換パラメータの変更は、限定された数のアプリケーションプログラムの種類によってのみ判断しており、必ずしもデータの種類を見て判断しているわけではない。このため、グラフィック主体のアプリケーションプログラムを使用して、多数のテキスト

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平9-319530

(43)公開日 平成9年(1997)12月12日

(51) Int.Cl.<sup>8</sup>

識別記号

室内整理番号

FI

### 技術表示簡所

G O 6 F 3/12

G 0 6 F 3/12

C

B 4 1 J 29/38

B 4 1 J 29/38

**Z**

審査請求 未請求 請求項の数 2 FD (全 16 頁)

(21)出願番号

特願平8-152912

(22) 出願日

平成8年(1996)5月27日

(71)出願人 000005496

富士ゼロックス株式会社

東京都港区赤坂二丁目17番22号

(72)発明者 藤本 厚史

神奈川県川崎市高津区坂戸3丁目2番1号

KSP R&amp;Dビジネスパークビル 富

士ゼロックス株式会社内

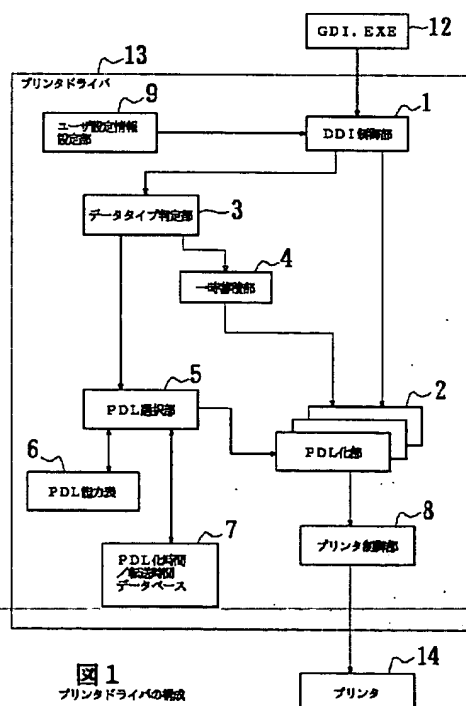
(74)代理人 弁理士 南野 貞男 (外1名)

(54) 【発明の名称】 プリンタドライバ

(57) 【要約】

【課題】 複数のページ記述言語に対応するプリンタを利用する場合に、プリンタドライバの側でデータ転送効率やデータ処理量および画像品質の点から最も適切なPDLを自動で選択することのできるプリンタドライバを提供する。

【解決手段】 プリントを制御するプリントドライバは、印刷データの特徴を判断する判断手段と、制御するプリンタで利用できる複数の種類のページ記述言語のうち最適なページ記述言語の種類を前記判断手段の判断結果に基づき選択する選択手段と、選択された種類のページ記述言語で前記印刷データを表現した出力データを作成する作成手段とを備える。選択手段は、過去の出力データの作成時間、前記出力データのプリンタへの転送時間を参照して、最適なページ記述言語を選択する。



**図 1**  
プリンタドライバの構造

## 【特許請求の範囲】

【請求項1】 印刷データの特徴を判断する判断手段

と、  
制御するプリンタで使用できる複数の種類のページ記述言語のうち最適なページ記述言語の種類を前記判断手段の判断結果に基づき選択する選択手段と、  
選択された種類のページ記述言語で前記印刷データを表現した出力データを作成する作成手段とを備えることを特徴とするプリンタドライバ。

【請求項2】 請求項1に記載のプリンタドライバにおいて、

前記選択手段は、過去の出力データの作成時間、前記出力データのプリンタへの転送時間を参照して、最適なページ記述言語を選択することを特徴とするプリンタドライバ。

## 【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、ページ記述言語を使用してプリントイメージを作成し印刷するプリンタのためのプリンタドライバに関し、特に、複数の種類のページ記述言語を解釈できるプリンタのためのプリンタドライバに関する。

【0002】

【従来の技術】昨今、プリンタの高機能化に伴って、複数のページ記述言語に対応できるプリンタが登場してきている。パーソナルコンピュータ上で動作するアプリケーションプログラムから、そのアプリケーションプログラムで作成した印刷データをプリンタに出力する場合を考えると、図6に示すように、プリンタ14が解釈するページ記述言語（以下、PDLと略称する）の描画命令は、パーソナルコンピュータ10とプリンタ14との間で受け渡しするための1種の画像情報の表現形式と見なすことができる。

【0003】PDLの描画命令を用いることにより、印刷する画像情報をそのままプリンタに転送する場合に比較して、データ転送量を減すことができる。また、標準化されたPDLを用いることにより印刷データの互換性を維持することができる。更に、プリンタが複数の種類のPDLの描画命令を受け入れる（解釈できる）ようにすることによって、パーソナルコンピュータ上のより広範なアプリケーションプログラムが生成する印刷データを扱えるようになる。

【0004】ところで、マイクロソフト社のオペレーティングシステム「WINDOWS」においては、図7に示すように、オペレーティングシステム「WINDOW S」上のアプリケーションプログラム11から送出される印刷データは、GDI(Graphics Device Interface)と呼ばれる共通のインタフェースのデータ形式により送出される。これをデバイスインタフェース処理部(GDI. EXE)12が受け取り、印刷データの画像情報を

プリンタドライバ13に渡す。この処理を行うデバイスインタフェース処理部(GDI. EXE)12は、「WINDOWS」オペレーティングシステムに含まれている共通のインタフェース処理機能である。つまり、デバイスインタフェース処理部(GDI. EXE)12が、アプリケーションプログラム11から送出される印刷データの画像情報を受け取り、プリンタドライバ13に適合するように形式変換して画像情報を渡す。プリンタドライバ13は、物理的に接続されているプリンタ14に対して、当該画像情報をそのプリンタに対応するPDLの描画命令や圧縮イメージに展開してプリンタ14に送る。プリンタドライバ13は、物理的に接続されるプリンタ14に適合するものを用意しておくことになる。

【0005】プリンタへのデータ転送時間を短縮するため、プリンタに印刷データを送る場合には、パーソナルコンピュータ内部で印刷データのデータ圧縮を行って一度蓄積してから送信する。このとき、例えば、特開平6-292024号公報に記載の「画像形成システム装置」のように、プリンタへのデータ転送時間を更に短縮するため、圧縮効率が悪いと判断される場合には、他の効率的な圧縮方法に当該印刷データのデータ圧縮方法を切り替えて圧縮する。このようにして、最適な圧縮方法を使用することによって、プリンタへの転送時間を短縮する。

【0006】この場合、プリンタに転送される画像情報の圧縮データを、パーソナルコンピュータとプリンタとの間で受け渡すための1種の画像情報の表現形式と見なすと、圧縮方法を選択することは、画像情報の1つの表現形式を選択するという意味と同様である。画像情報の1つの表現形式を選択するという意味では、マルチPDL対応のプリンタにおいて、印刷データの表現形式の中の複数のPDLの形式から1つのPDLの形式を選択することと等価であると見なせる。

【0007】ところで、複数のPDLに対応できるマルチPDL対応プリンタの場合、PDLごとに別々のプリンタドライバが提供されるのが通例であるので、上記の技術を応用する場合、プリンタへのデータ転送時間を短縮するために、PDLによるデータ量の圧縮を評価（判断）して、適切なPDLを用いるプリンタドライバを選択することになる。

【0008】また、プリンタドライバを選択する技術に関しては、例えば、文献「“HP Bids Farewell to 300 dpi with New 4 ppm LaserJet 5L” THE HARD COPY OBSERVER, Oct. 1995」に記載されている内容が参照できる。この文献によると、プリンタに対応して、そのプリンタドライバをインストールする際には、プリンタドライバが動作する環境に応じて、例えば、オペレーティングシステムがマイクロソフト社の「DOS」なのか「WINDOWS」なのかに対応して、また、CPUの速度の大小、搭載しているメモリ量の大小などに応じて、イ

インストールするプリンタドライバを選択する技術が紹介されている。例えば、オペレーティングシステムが「WINDOWS」で、高速なCPUを使用し、多くのメモリを搭載してあるシステムの場合には、処理するデータ量のより多い「Windows Printing System」用のプリンタドライバをインストールし、オペレーティングシステムが「DOS」環境であれば、処理するデータ量の少ないPDL用のドライバをインストールする。

【0009】また、同文献によれば、アプリケーションプログラムの種類を認識し、そのアプリケーションプログラムによってプリンタドライバの動作パラメータを自動的に変更する技術についても紹介されている。これによると、例えば、マイクロソフト社の商品名「Microsoft Power Point」のようなグラフィックスを中心に扱うアプリケーションプログラムの場合は、グラフィックスの再現性を重視するパラメータを選択し、マイクロソフト社の商品名「Microsoft Word」のようにテキストを中心に扱うアプリケーションプログラムの場合は、テキストの再現性を重視するパラメータを選択する。

【0010】

【発明が解決しようとする課題】ところで、前述の特開平6-292024号公報に記載の「画像形成システム装置」による技術をプリンタドライバに適用する場合においては、プリンタ側に転送する画像データを実際に圧縮してみないと、どの圧縮方法が圧縮効率が高いかわからない。このため、何回も圧縮しなければならない場合があり、その変換に多くの時間がかかるという問題がある。

【0011】また、この特開平6-292024号公報に記載の「画像形成システム装置」の技術におけるデータ圧縮によるデータの表現形式の変換を、PDLの描画命令への変換(PDL化)に読み替えて、マルチPDL対応プリンタのプリンタドライバにおける最適なPDLの種類を選択することに適用すると、その場合、PDL自身の表現能力に差があるため、単純に圧縮効率だけによってPDLの種類を選択すると、画質が劣化する場合があるという問題が生じる。

【0012】前述したように、アプリケーションプログラム11(図7)から出力される印刷データによる画像情報(描画コマンド)は、デバイスインタフェース処理部(GDI.EXE)12が受け取り、プリンタドライバ13に適合するような形式に変換して渡すので、プリンタドライバ13が、接続されているプリンタ用のPDL命令や圧縮イメージに展開して、プリンタ14に送る処理を行う。その場合、アプリケーションプログラム11からは、デバイスインタフェース処理部(GDI.EXE)12を介して、プリンタドライバ13に描画コマンドを発行するので、プリンタドライバ13は、デバイスインタフェース処理部(GDI.EXE)12からの描画の指示を受け取り、その指示をPDL命令に対応づ

けることになる。

【0013】その場合、プリンタに対応したプリンタドライバ13において、使用するPDLの描画処理の機能が十分であれば、図8に示すように、動作開始時に、プリンタドライバ13では、デバイスインタフェース処理部(GDI.EXE)12から描画機能の問い合わせコマンド(Enable)があると、これに対して、その能力があることを宣言する(GDI.INFO)ので、これにより、例えば、アプリケーションプログラム11からの楕円描画命令「Ellipse」がデバイスインタフェース処理部(GDI.EXE)12を介して、そのままの形式「OUTPUT(Ellipse)」プリンタドライバ13に伝え、プリンタドライバ13から楕円描画のためのPDL命令(楕円描画)が出力される。

【0014】しかし、プリンタドライバのPDLの描画処理の機能が十分でない場合には、図9に示すように、動作開始時に、プリンタドライバ13は、デバイスインタフェース処理部(GDI.EXE)12から描画機能の問い合わせコマンド(Enable)があっても、その能力がないことを宣言する(GDI.INFO)ことになるので、例えば、アプリケーションプログラム11から送出された楕円描画命令は、デバイスインタフェース処理部(GDI.EXE)12によって複数のライン描画命令「OUTPUT(SCANLINE)」に置き換えられて描画指示される。この場合、プリンタドライバ13は、そのライン描画命令「OUTPUT(SCANLINE)」からのPDL命令への置き換えは、必ずしも接続されているプリンタに適切な形で分解されないため、もともとPDL命令によって楕円描画がサポートされている場合に比較して描画品質が大きく劣化する。

【0015】また、上記の文献に記載の技術のように、インストールする際にシステム的环境に応じて、プリンタドライバを選択する方式を用いると、インストール時には確かに動作環境に応じたPDLの選択は可能であるが、インストール後は常に同一のプリンタドライバが使用されるため、印刷データに応じた動的なPDLの選択は行えないという問題がある。

【0016】また、同じく、上記文献に記載の技術のうち、アプリケーションプログラムによって動作パラメータを変更する方式によると、1種類のPDLへの変換パラメータをアプリケーションプログラムに応じて変えているだけであるため、その場合、プリンタ側に複数のPDL生成部が実装されていても、それら複数のPDLを選択的に使用できず、印刷データに適切なPDLの選択までは行えないという問題がある。

【0017】ここでは、変換パラメータの変更は、限定された数のアプリケーションプログラムの種類によってのみ判断しており、必ずしもデータの種別を見て判断しているわけではない。このため、グラフィック主体のアプリケーションプログラムを使用して、多数のテキスト



を含んだ文書を印刷した場合には、パラメータ変更の効果が必ずしも得られないばかりか、場合によっては、何も処理を施さない場合よりも悪化する可能性があるという問題がある。

【0018】本発明は、これらの問題を解決するためになされたものである。本発明の目的は、プリンタドライバにおいて、複数のPDLに対応するプリンタを利用する場合に、プリンタドライバの側でデータ転送効率やデータ処理量および画像品質の点から最も適切なPDLを自動で選択することのできるプリンタドライバを提供することにある。

【0019】

【課題を解するための手段】上記のような目的を達成するため、本発明のプリンタドライバは、第1の特徴として、印刷データの特徴を判断する判断手段(3)と、制御するプリンタで利用できる複数の種類のページ記述言語のうち最適なページ記述言語の種類を前記判断手段の判断結果に基づき選択する選択手段(5)と、選択された種類のページ記述言語で前記印刷データを表現した出力データを作成する作成手段(2)とを備えることを特徴とする。

【0020】また、本発明のプリンタドライバは、第2特徴として、前記選択手段が、過去の前記出力データの作成時間、前記出力データのプリンタへの転送時間を参照して、最適なページ記述言語の種類を選択することを特徴とする。

【0021】このような特徴を有する本発明のプリンタドライバによれば、複数のページ記述言語に対応するプリンタを利用する場合に、プリンタドライバの側でデータ転送効率やデータ処理量および画像品質の点から最も適切なページ記述言語の種類が自動で選択される。その選択基準は、例えば、マルチPDLプリンタに出力時に、過去のPDL化に要した時間やデータ転送量などの統計情報とユーザ指定の条件から原稿の印刷データに対して適切なPDLを採用する。つまり、この選択基準では、次の3つの条件により印刷データに適切なPDLを選択する。

- (1) ユーザ所望の描画品質を満たす。
- (2) PDL化時間が短い。
- (3) データ転送時間が短い。

このような選択基準を選択手段が保持し、選択手段が保持している選択基準の参照によって、最適なPDLを選択することにより、プリンタドライバでは、マルチPDL対応のプリンタにおいて印刷データに対して最も適した種類のPDLを利用してプリンタに送信することができ、最大の描画品質を保ちながら、かつPDL化時間やデータ転送時間を最小にすることができる。

【0022】

【発明の実施の形態】以下、本発明を実施する場合の形態について、図面を参照して具体的に説明する。図1

は、本発明の一実施例のプリンタドライバの要部の構成を示すブロック図である。図1において、1はデバイスドライバインタフェース制御部(DDI制御部)、2はページ記述言語命令生成部(PDL化部)、3はデータタイプ判定部、4は一時蓄積部、5はページ記述言語選択部(PDL選択部)、6はページ記述言語能力表(PDL能力表)、7はPDL化時間/データ転送時間データベース、8はプリンタ制御部、9はユーザ設定情報設定部、12はデバイスインタフェース処理部、13はプリンタドライバ、14はプリンタである。

【0023】この実施例では、「WINDOWS」オペレーティングシステム上で動作するプリンタドライバを例として説明する。「WINDOWS」オペレーティングシステム上で動作するアプリケーションプログラムは、印刷データをプリンタドライバに出力する場合、「WINDOWS」オペレーティングシステムのインタフェースプログラムを介して印刷データを送出する。その場合、印刷データは、グラフィックス・デバイス・インタフェース(GDI)形式により、デバイスインタフェース処理部(GDI.EXE)12に渡され、更に、デバイスインタフェース処理部(GDI.EXE)12によってデータ形式の変換がなされて、プリンタドライバ13に渡される。

【0024】プリンタドライバ13においては、図1に示すように、デバイスインタフェース処理部(GDI.EXE)12から渡された印刷データを、デバイスドライバインタフェース制御部(以下、DDI制御部と略称する)1を介して受け取り、ページ記述言語命令生成部(以下、PDL化部と略称する)2に送出すると共にデータタイプ判定部3にも送出する。

【0025】この場合、まず、DDI制御部1の制御処理では、動作開始時に、まず、ユーザ設定情報設定部9に保持されているユーザ設定情報を使って、デバイスインタフェース処理部(GDI.EXE)12に対し、プリンタドライバとしての能力(PDL生成能力)を宣言する。この能力の宣言は、例えば、プリンタドライバの能力に応じてデバイスインタフェース処理部(GDI.EXE)12が発行する描画コマンドを制限したい時に使用する。

【0026】また、DDI制御部1は、ユーザ設定情報のPDL指定が「自動選択」以外に設定されている場合には、使用するPDLの種類が既に指定されているので、その対応のPDL化部2に対して、渡された印刷データを直接に送出する。ユーザ設定情報のPDL指定が「自動選択」になっている場合には、渡された印刷データは、データタイプ判定部3に渡し一時蓄積部4において一時蓄積する。

【0027】データタイプ判定部3では、受け取った印刷データに含まれているテキスト、グラフィックス、イメージの割合を調べて、データタイプを決定する。決定

したデータタイプは、ページ記述言語選択部（以下、PDL選択部と略称する）5に送出する。なお、この時の印刷データ自体は、一時蓄積部4において一時蓄積される。

【0028】データタイプ判定部3において判定されたデータタイプが、PDL選択部5に伝えられると、PDL選択部5では、ページ記述言語能力表（以下、PDL能力表と略称する）6を参照し、受け取った印刷データを表現するのに十分な能力を持ったPDLの種類を選択する。ここで1つの特定のPDLの種類に決定できれば、決定したPDLの種類をPDL化部2に伝えて、決定した種類のPDLの描画命令の生成処理（PDL化処理）を指示する。

【0029】また、PDL選択部5では、選択すべきPDLの種類候補として複数残った場合、PDL化時間／転送時間データベース7を参照し、それぞれの複数の候補のPDLの種類について、印刷データのデータタイプに対する過去の実績データ（統計情報）からPDL化の予測時間とデータ転送の予測時間の和の推定時間を求め、この推定時間が最小となるPDLの種類候補を選択し、PDLの種類を決定する。そして、決定したPDLの種類をPDL化部2に伝え、PDL化処理を指示する。

【0030】PDL化部2では、指示された種類のPDLにより、印刷データに対応するそれぞれの描画命令を生成し、プリンタ制御部8に出力する。プリンタ制御部8では、受け取ったPDLの描画命令をプリンタ装置14に送出する。

【0031】図2は、プリンタドライバによる制御処理の一連の動作を示すフローチャートである。すなわち、プリンタドライバによる制御処理の動作は、図2に示すように、デバイスドライバインタフェース（DDI）によりプリントデータを受け取って、制御処理の動作を開始する（ステップ21）と、プリントデータのデータタイプを判定する（ステップ22）。この判定結果により、プリントデータの中にグラフィックスやイメージを含むか否かを判定する（ステップ23）。グラフィックスやイメージを含む場合には、PDL能力表で直接対応づかない描画命令数を計数し、最小のPDLを選択する（ステップ24）。その選択によりPDLが1つ決まるか否かを判定する（ステップ25）。

【0032】PDLが1つに決まらない場合や、グラフィックスやイメージを含まない場合には、統計データを参照してPDL化予測時間と転送予測時間の和が最小のPDLを選択する（ステップ26）。これにより、最適なPDLを決定できるので、ここで選択されたPDLにより、印刷するデータをPDL化し、プリンタに送信する（ステップ27）。

【0033】次に、このようなプリンタドライバを構成する要素（各モジュール）について更に説明する。ユー

ザ設定情報設定部9は、デバイスインタフェース処理部（GDI、EXE）12への能力宣言情報と、PDL指定情報を設定する。

【0034】ここでのデバイスインタフェース処理部（GDI、EXE）12への能力宣言情報は、デバイスインタフェース処理部（GDI、EXE）12から発行される描画コマンドを制限したい時に指定する。通常の場合の指定は「全ての能力」が指定される（規定値）。

【0035】また、デバイスインタフェース処理部（GDI、EXE）12から発行された描画コマンドに対応する描画機能（PDL命令）が、最終的に決定（選択）された種類のPDLに存在しない場合、通常の場合にはその都度エラーを返し、デバイスインタフェース処理部（GDI、EXE）12に対して適切なブレークダウン（描画コマンドの変更）を依頼する。

【0036】PDL指定情報は、特定のPDLを使用したいときに、その特定のPDLを指定するために用いる。PDL指定情報としては、通常の場合は「自動選択」を指定しておくが、画質や再現性を重視して特定のPDLを使用するように指定することもできる。例えば、中間調を美しく表現したい場合には、PDLの種類として“PostScript”を指定しておく。その場合、特定のPDLの指定がなされていると、自動選択の機能はバイパスされ、DDI制御部1の制御処理において、対応するPDLの種類をPDL化部2に直接に印刷データを渡すようになる。

【0037】前述したように、データタイプ判定部3では、受け取ったプリントデータのテキスト、グラフィック、イメージの割合を調べて、データタイプを決定する。すなわち、このデータタイプを決定では、DDI制御部1から渡される描画コマンドが、テキスト描画コマンド、グラフィックス描画コマンド、イメージ描画コマンドのいずれかに分類することができるので、それぞれの種類の描画コマンドのコマンド数を調べることで、データタイプを決定する。その結果、データタイプは、次のいずれかに分類されることになる。

- (1)．テキストのみのタイプ、(2)．グラフィックスのみのタイプ、(3)．イメージのみのタイプ、
- (4)．テキストとグラフィックスのみのタイプ、
- (5)．テキストとイメージのみのタイプ、(6)．グラフィックスとイメージのみのタイプ、(7)．テキストとグラフィックスとイメージを全て含むのタイプ。

【0038】PDL能力表6は、デバイスインタフェース処理部（GDI、EXE）12からの描画コマンドをPDLの描画命令に置換する際に、画質劣化を招くような描画命令がどれぐらい使われているかを判定するために使用される。

【0039】図3は、PDL能力表の一例を示す図である。図3に示すように、PDL能力表6には、各々のPDLの種類毎に、デバイスインタフェース処理部（GDI

I. EXE) 12から発行される各々の描画コマンドに当該PDLの描画命令が対応できるか否かが示されている。具体例で説明すると、この場合、デバイスインタフェース処理部(GDI)が発行される描画コマンドが曲線描画命令の「Output (ARC)」である場合には、第1番目のPDLの「PDL#1」、第2番目のPDLの「PDL#2」、第3番目のPDLの「PDL#3」のいずれの種類のPDLでも対応可能であるが、描画命令が楕円描画命令の「Output (ELLIPSE)」である場合には、第1番目のPDLの「PDL#1」は対応可能であるが、第2番目のPDLの「PDL#2」、第3番目のPDLの「PDL#3」のいずれも対応不可能である。また、描画命令が円描画命令の「Output (CIRCLE)」である場合には、第1番目のPDLの「PDL#1」と第2番目のPDLの「PDL#2」は対応可能であるが、第3番目のPDLの「PDL#3」は対応不可能である。

【0040】このようなPDL能力表6を参照することにより、現在の印刷データについて、デバイスインタフェース処理部(GDI. EXE) 12から発行される描画コマンドが、どれ位の割合でPDLの描画命令に置換されるかが判定できる。

【0041】次に、PDL化時間/転送時間データベース7について説明する。図4はPDL化時間/転送時間データベース7の内容を説明する図である。図4に示すように、PDL化時間/転送時間データベース7は、原稿タイプフィールド7a、PDL種類フィールド7b、PDL化時間フィールド7c、PDLデータ量フィールド7d、およびサンプル数フィールド7eから構成されており、各々のフィールドのデータとして、原稿のタイプ(データタイプ)に応じて、各々のPDLの種類毎に、そのPDLを使用した場合のPDL化時間(原稿1MBあたり)、PDLデータ量(原稿1MBあたり)のそれぞれのデータが、サンプル数の統計データと共に格納されている。

【0042】具体例で説明すると、PDL化時間/転送時間データベース7の第1行目のデータ(レコードデータ)は『原稿のタイプが「テキストのみ」で、PDLの種類が「PDL#1」のPDLを用いた場合に、そのPDL化時間が、原稿1MBあたり「5秒」であり、PDLデータ量は、原稿1MBあたり「1.2MB」となる。このデータは、過去125回の印刷データのサンプルから得られたものである』ことを意味している。

【0043】印刷データのタイプが、グラフィックスやイメージを含まない場合、また、直接対応づかない描画命令の数が同等の場合などにおいては、前述のPDL能力表6を参照しても、PDLの種類を選択ができないので、PDL化時間/転送時間データベース7に格納されている統計データ(レコードデータ)を用いて、PDL化予測時間と転送予測時間の和が最小となるPDLの種

類を選択するために用いる。

【0044】PDL化時間/転送時間データベース7のデータ内容は、過去の印刷データの統計データにより作成される。これにより、過去の統計データから経験的にその信頼度を高くして、印刷データに対するPDL化予測時間と転送予測時間を予測し、これらの予測時間の和が最小のPDLの種類を選択する。このときの転送予測時間は、出力予定のポートの転送速度とPDLのデータ量との関係から次式により計算する。

$$(\text{転送予測時間}) = (\text{生成されるPDLデータ予測量}) / (\text{出力ポートのデータ転送速度})$$

なお、全てのPDLについての印刷データの統計データが揃うまでのPDLの自動選択は、例えば、ラウンドロビン法によって決定する。これにより、全てのPDLに対してデータを収集することができる。

【0045】PDL化時間/転送時間データベース7におけるデータの更新は、プリント要求に対する処理を終了することに行われる。これは、つまり、PDLを自動選択せず、特定のPDLを使用するように指定された場合でもデータは更新されることを意味する。データの更新は、次のように行う。現在までの統計値について、PDL化時間をAとし、PDLのデータ量をBとし、サンプル数をnとし、今回のプリントにおける実測値について、PDL化時間をXとし、PDLのデータ量をYとし、今回のプリント後の統計値については、PDL化時間をA'とし、PDLのデータ量をB'とし、サンプル数をn'とすると、

$$A' = (A \times n + X) / (n + 1)$$

$$B' = (B \times n + Y) / (n + 1)$$

$$n' = n + 1$$

として、PDL化時間/転送時間データベース7における統計データ(レコードデータ)の更新を行う。

【0046】このようにして、マルチPDL対応のプリンタに出力する場合において、過去の印刷データからの統計データを用いて、プリントデータに最も適したPDLの種類を選択し、選択したPDLによる描画命令を利用してプリンタに送信できるようになるので、最大の描画品質を保ちながら、なおかつPDL化時間や転送時間を最小にすることができる。

【0047】つまり、本発明によるプリンタドライバによれば、当該プリンタドライバに渡されたデータの種類を見分け、その識別するデータの種類をテキスト、グラフィックス、イメージの割合で決め、グラフィックスやイメージを含む場合には、更に、PDL能力表6を参照して、直接に対応づかない描画命令が、一番少ないPDLを選択する。また、グラフィックスやイメージを含まない場合や、直接に対応づかない描画命令の数が同等の場合には、PDL化時間/転送時間データベース7における過去の統計データを参照し、PDL化の予測時間と、転送の予測時間の和が最小となるPDLを選択す

る。そして、選択したPDL命令を用いて印刷データをその描画命令に変換して、プリンタに送出する。これにより、最大の描画品質を保ちながら、なおかつPDL化時間および転送時間を最小にすることができる。

【0048】次に、上記のように構成される本実施例のプリンタドライバによる処理について、全体の処理の流れを処理フローを参照して具体的に説明する。図5は、本発明の実施例のプリンタドライバによる処理フローを示すフローチャートである。プリンタドライバの処理では、処理を開始すると、まず、ステップ51において、ユーザ設定情報を使って、デバイスインタフェース「GDI. EXE」に描画能力を宣言する。次に、ステップ52において、デバイスドライバインタフェース(DDI)によりプリントデータを受け取る。次のステップ53において、PDL指定が「自動選択」であるか否かを判断する。

【0049】PDL指定が「自動選択」でない場合、プリントデータに使用するPDLの種類がユーザにより予め指定されているので、直ちにステップ59に進み、指定の種類のPDLを用いて、プリントデータをPDL化してプリンタに送信する。一方、ステップ53の判定において、PDL指定が「自動選択」である場合には、プリントデータの特徴から最適なPDLの種類を判定するための処理を行う。このため、次のステップ54において、プリントデータのデータタイプを判定し、そのプリントデータは一時蓄積する。次に、ステップ55において、データタイプの判定したプリントデータの中にグラフィックスやイメージを含むか否かを判別し、グラフィックスやイメージを含む場合には、次のステップ56に進み、PDL能力表で直接対応づかない描画命令数を計数し、最小のPDLを選択する。そして、次のステップ57において、その選択によりPDLが1つ決まるか否かを判定する。PDLが1つに決まらない場合には、次のステップ58に進む。

【0050】一方、ステップ55の判定において、グラフィックスやイメージを含まない場合にも、ステップ58に進む。ステップ58においては、統計データを参照してPDL化予測時間と転送予測時間の和が最小のPDLを選択する。これにより、最適なPDLを決定できるので、次のステップ59において、ここで選択されたPDLにより、印刷するデータをPDL化し、プリンタに送信して、ここでの処理を終了する。

【0051】ところで、上述した実施例においては、印刷データからデータの種類の見分ける場合、そのデータタイプを「①テキストのみ、②グラフィックスのみ、③イメージのみ、④テキストとグラフィックスのみ、⑤テキストとイメージのみ、⑥グラフィックスとイメージのみ、⑦テキストとグラフィックスとイメージを全て含む」の7種類に分類しているが、この場合において、例えば、同じ「④テキストとグラフィックスのみ」の場合

でもグラフィックスの率の大小によって、選択すべき最適なPDLが異なる場合があり得る。

【0052】これに対しては、例えば、PDL化時間／転送時間データベース(7:図4)のデータの中で、原稿タイプとして、更に詳細にその印刷データの種別(データタイプ)を反映させたデータ内容とする。図10は、PDL化時間／転送時間データベースの別の一例を説明する図である。この場合、更に、印刷データの描画コマンドにおけるテキスト描画コマンドの数とグラフィックス描画コマンドの数との比を考慮し、図10に示すように、原稿タイプの種類として、グラフィックスコマンドの比率が50%の上下によって2種類に分離したレコードデータを設ける。これにより、前記の実施例の効果に加えて、より原稿の特性に合わせたPDLを選択できるようになる。

【0053】具体例で説明する。図9に示すように、PDL化時間／転送時間データベース70においては、原稿タイプの種類として「テキストとグラフィックスのみ」については、「テキストとグラフィックスのみ(グラフィックスは50%未満)」と「テキストとグラフィックスのみ(グラフィックスは50%以上)」の2種類に分けて、それぞれのレコードデータを設け、原稿タイプの種類に応じてそれぞれ印刷データの統計データを更新している。

【0054】この結果、印刷データのタイプとして「テキストとグラフィックスのみ」であっても、その中のグラフィックス描画コマンドが50%以上である場合には、PDLの種類を「PDL#1」とすると、そのPDL化時間が、原稿1MBあたり「2秒」で済むという統計データが得られる。また、グラフィックス描画コマンドが50%未満である場合に、PDLの種類を「PDL#1」とすると、そのPDL化時間が、原稿1MBあたり「6秒」かかるという統計データが得られることになる。これらの統計データをPDL化時間／転送時間データベース70に反映させて、このような統計データを用いて、PDLを選択することにより、より原稿の特性に合わせたPDLを選択できるようになる。

【0055】更に、別の変形例について説明する。前述の実施例においては、印刷データから原稿タイプを判定する場合に、文書全体を一つの処理単位として扱っているので、このため、1ページ目がテキストだけで、2ページ目がイメージだけというような2ページの文書では、文書タイプが「テキストとイメージだけ」というデータタイプに判定されてしまう。この判定による基準によって、適切なPDLを選択しようとしても、最適なPDLの種類を選択できない場合がある。

【0056】これに対しては、つまり、ページ毎に印刷データの画像の特徴が明確に分かれている文書が多い場合に対しては、ページ毎に別の種類のPDLを使用するように構成する。これにより、より効率のよいプリント

処理を行うことができる。

【0057】このためには、前記の実施例の処理において、1つのドキュメントをページと読み替えるだけでよい。この場合、現状のプリント処理の仕組みの中では、1ドキュメントが1ジョブになるので、ページ毎に別のジョブとして扱う必要がある。このため、各々のジョブ毎にバナー（見出し）シートなどを付ける設定がなされていると、PDLが変更されるごとにバナーシートが出力されてしまうため、バナーシートの出力は行わないような設定にする。

【0058】ところで、DPA（Document Printing Application; ISO10175）にしたがったプリンタの場合、1ジョブで複数のドキュメントが扱えるオプションがあるので、このようなプリンタの場合にはページ毎に別々のドキュメントとしても、全体は1つのジョブとみなせることため、上記のような制限は緩和される。

【0059】これにより、マルチPDLプリンタに出力する際、ページごとのプリンタデータに最も適したPDLを利用してプリンタに送信することができるので、最大の描画品質を保ちながら、なおかつPDL化時間や転送時間を最小にすることがより効率よくできるようになる。

【図面の簡単な説明】

【図1】 図1は本発明の一実施例のプリンタドライバの要部の構成を示すブロック図、

【図2】 図2はプリンタドライバによる制御処理の一連の動作を示すフローチャート、

【図3】 図3はPDL能力表の一例を示す図、

【図4】 図4はPDL化時間／転送時間データベースを説明する図、

【図5】 図5は本発明の一実施例のプリンタドライバによる処理フローを示すフローチャート、

【図6】 図6はパーソナルコンピュータとプリンタの関係を説明する図、

【図7】 図7はプリンタドライバの位置付けを説明する図、

【図8】 図8はデバイスインタフェース処理部による描画命令の変換の第1の例を説明する図、

【図9】 図9はデバイスインタフェース処理部による描画命令の変換の第2の例を説明する図、

【図10】 図10はPDL化時間／転送時間データベースの別の一例を説明する図、

【符号の説明】

1…デバイスドライバインタフェース制御部（DDI制御部）、2…ページ記述言語命令生成部（PDL化部）、3…データタイプ判定部、4…一時蓄積部、5…ページ記述言語選択部（PDL選択部）、6…ページ記述言語能力表（PDL能力表）、7…PDL化時間／データ転送時間データベース、8…プリンタ制御部、9…ユーザ設定情報設定部、12…デバイスインタフェース処理部、13…プリンタドライバ、14…プリンタ。

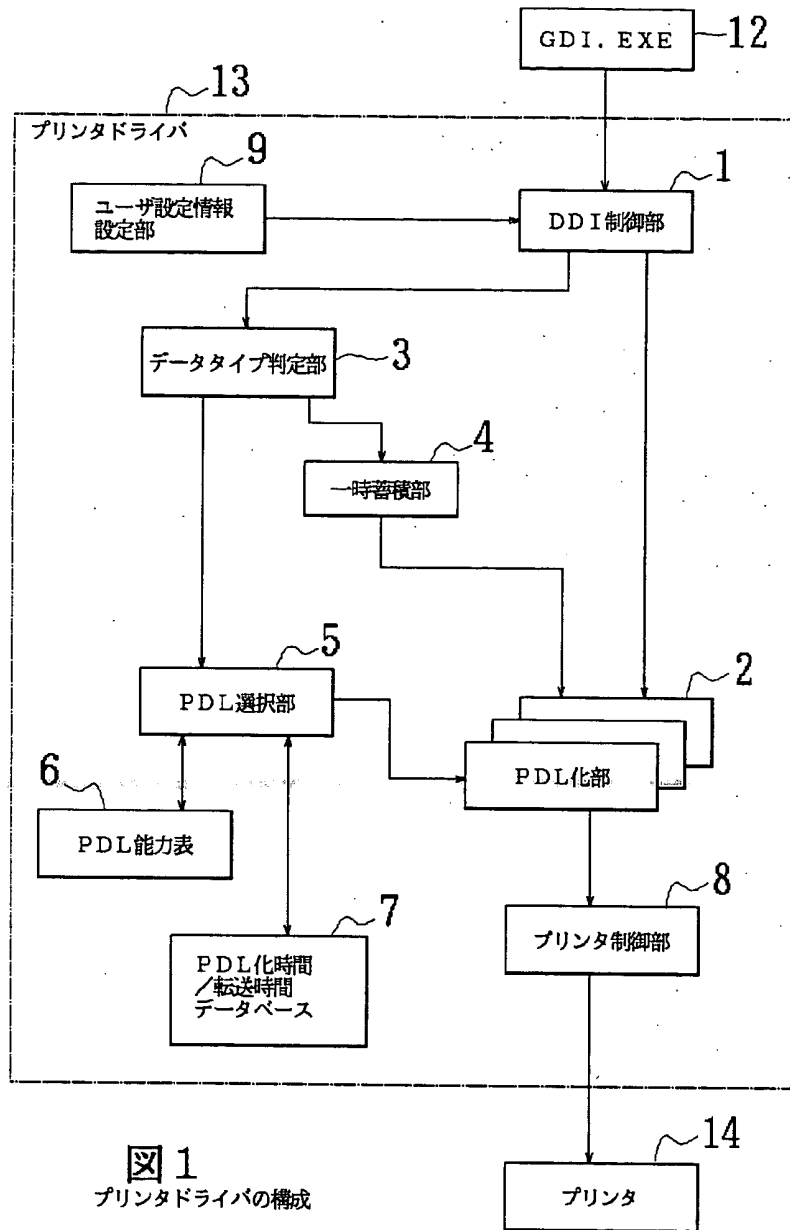
【図3】

6

GDIが発行する描画コマンド	PDL#1	PDL#2	PDL#3
Output (ARC)	○	○	○
Output (ELLIPSE)	○	×	×
Output (CIRCLE)	○	○	×
BitBlt	○	×	×
.....	.....	.....	.....
.....	.....	.....	.....

図3 PDL能力表

【図1】



【図2】

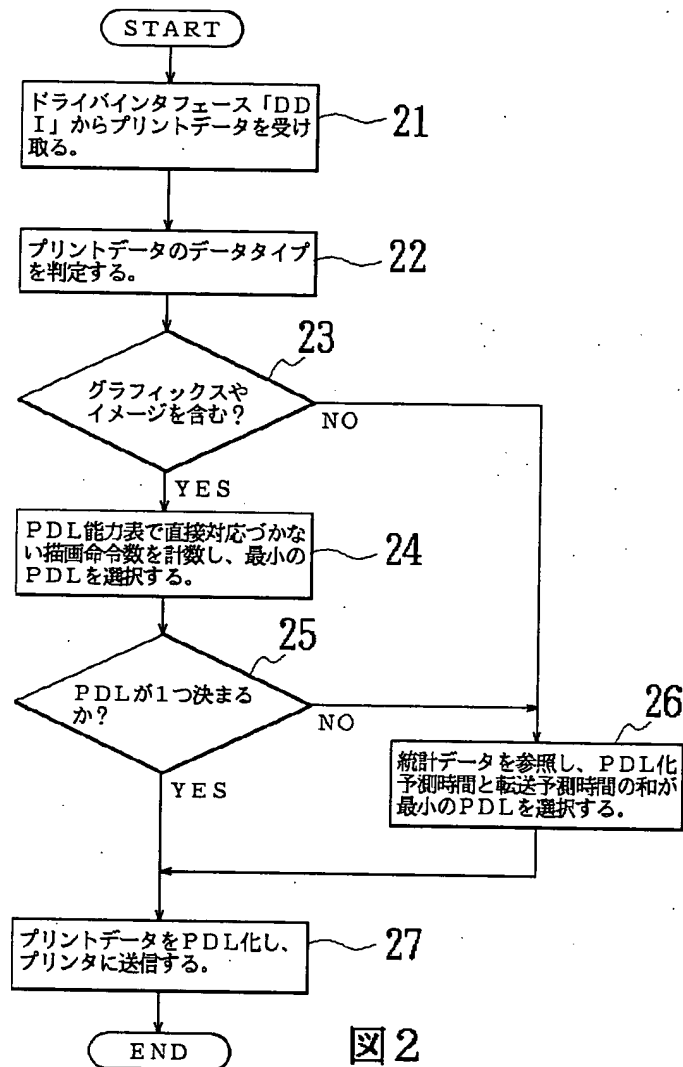


図2

【図6】

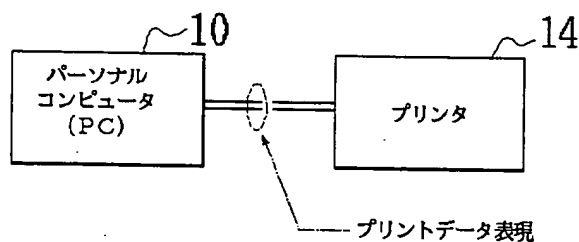


図6 パーソナコンピュータとプリンタの関係

【図4】

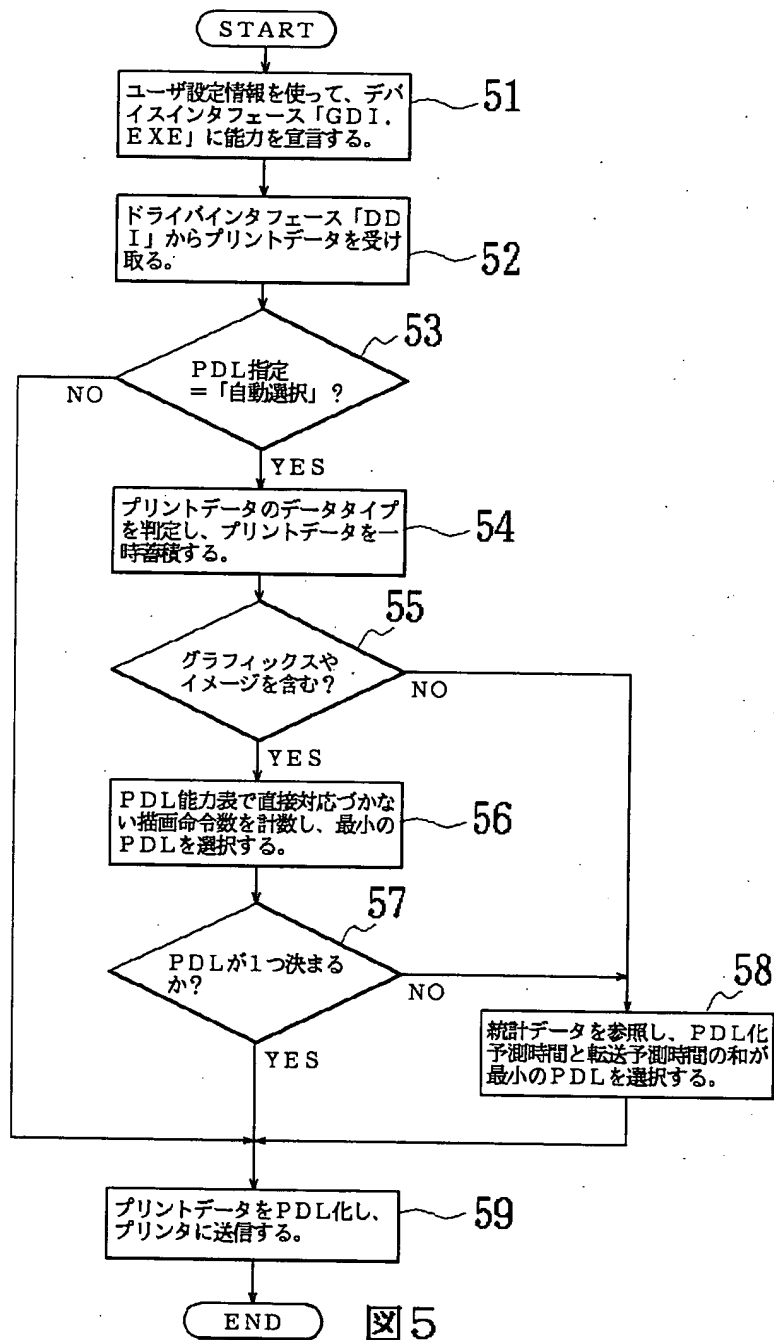
原稿タイプ	PDL #	PDL化時間 (原稿1MBあたり)	PDLデータ量 (原稿1MBあたり)	サンプル数
テキストのみ	PDL # 1	5.0 sec/1MB	1.2 MB/1MB	125
	PDL # 2	3.5 sec/1MB	1.1 MB/1MB	526
	PDL # 3	5.5 sec/1MB	1.1 MB/1MB	21
グラフィックスのみ	PDL # 1	6.5 sec/1MB	1.1 MB/1MB	32
	PDL # 2	3.5 sec/1MB	2.1 MB/1MB	658
	PDL # 3	4.0 sec/1MB	1.5 MB/1MB	35
イメージのみ	PDL # 1	7.5 sec/1MB	1.2 MB/1MB	6
	PDL # 2	9.0 sec/1MB	1.1 MB/1MB	325
	PDL # 3	2.0 sec/1MB	1.2 MB/1MB	54
テキストとグラフィックスのみ	PDL # 1	6.0 sec/1MB	1.1 MB/1MB	14
	PDL # 2	7.0 sec/1MB	1.8 MB/1MB	78
	PDL # 3	9.0 sec/1MB	1.3 MB/1MB	852
...	...	...	...	...
...	...	...	...	...

図4

PDL化時間／転送時間データベース



【図5】



【図7】

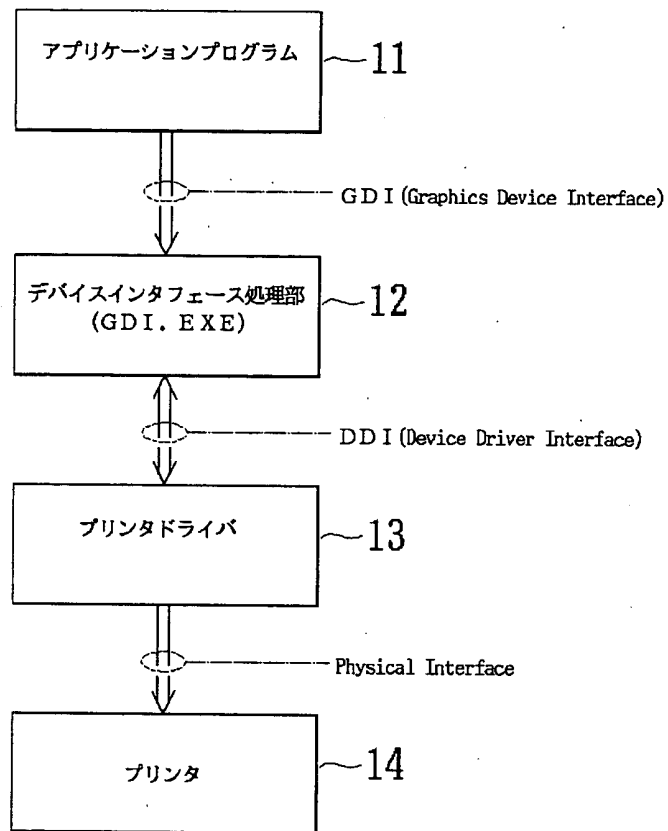


図7 プリンタドライバの位置付け

【図8】

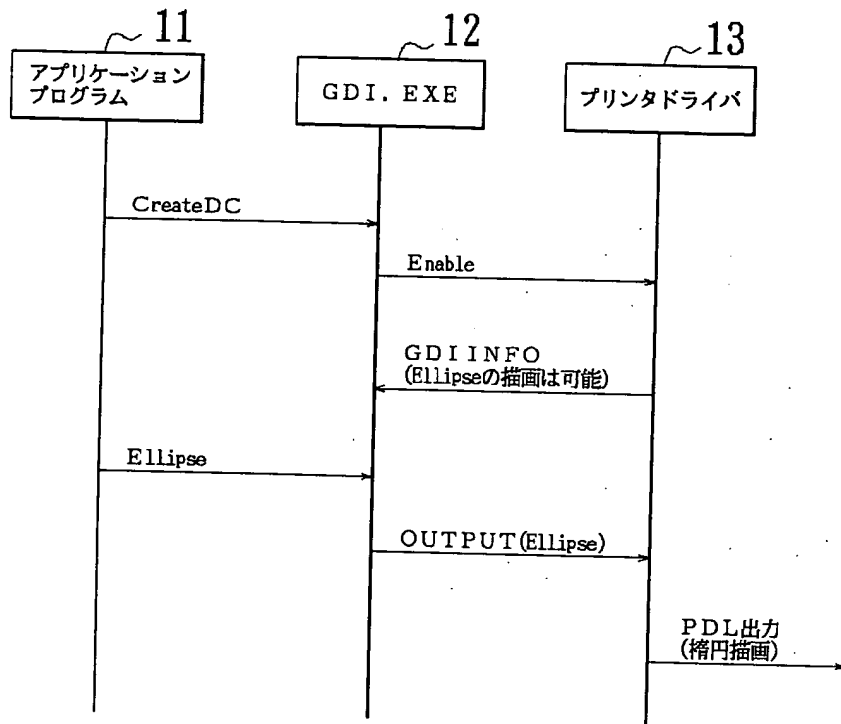


図8 描画命令の変換 (その1)

【図9】

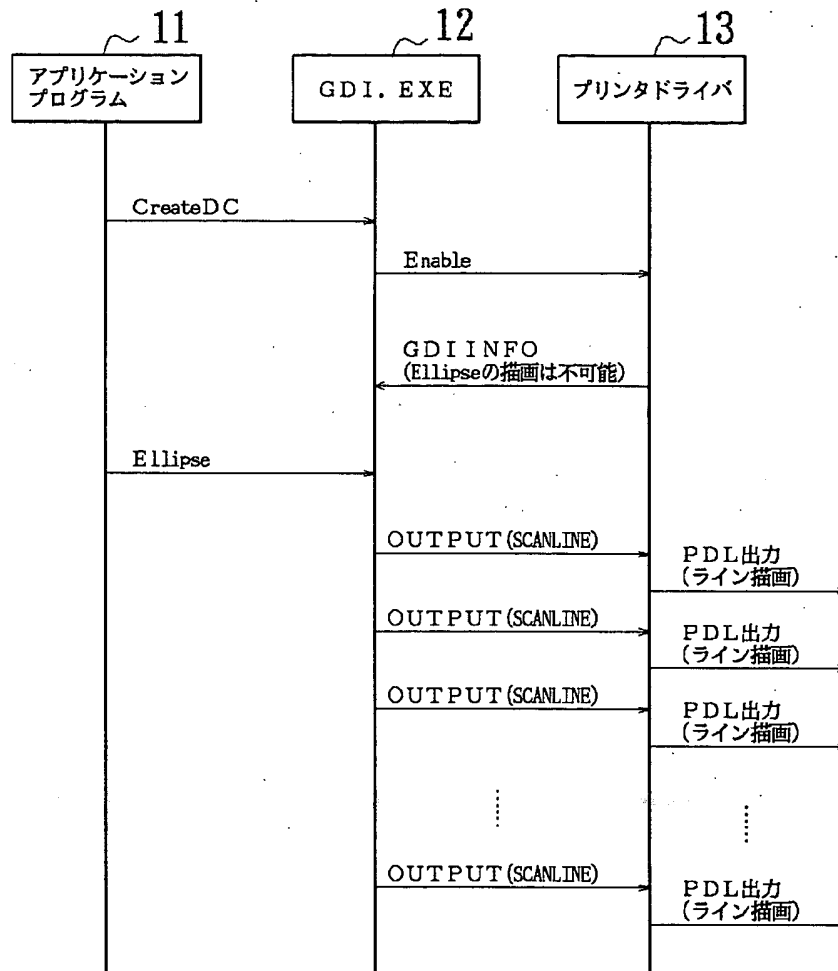


図9 描画命令の変換 (その2)

【図10】

70

原稿タイプ	PDL #	PDL化時間 (原稿1MBあたり)	PDLデータ量 (原稿1MBあたり)	サンプル数
テキストのみ	PDL # 1	5.0 sec/1MB	1.2 MB/1MB	125
	PDL # 2	3.5 sec/1MB	1.1 MB/1MB	526
	PDL # 3	5.5 sec/1MB	1.1 MB/1MB	21
グラフィックスのみ	PDL # 1	6.5 sec/1MB	1.1 MB/1MB	32
	PDL # 2	3.5 sec/1MB	2.1 MB/1MB	658
	PDL # 3	4.0 sec/1MB	1.5 MB/1MB	35
イメージのみ	PDL # 1	7.5 sec/1MB	1.2 MB/1MB	6
	PDL # 2	9.0 sec/1MB	1.1 MB/1MB	325
	PDL # 3	2.0 sec/1MB	1.2 MB/1MB	54
テキストとグラフィックスのみ (グラフィックスは50%未満)	PDL # 1	6.0 sec/1MB	1.1 MB/1MB	14
	PDL # 2	7.0 sec/1MB	1.8 MB/1MB	78
	PDL # 3	9.0 sec/1MB	1.3 MB/1MB	852
テキストとグラフィックスのみ (グラフィックスは50%以上)	PDL # 1	2.0 sec/1MB	1.1 MB/1MB	58
	PDL # 2	4.0 sec/1MB	1.1 MB/1MB	32
	PDL # 3	6.0 sec/1MB	1.1 MB/1MB	258
...	...	...	...	...
	...	...	...	...

図10

**This Page Blank (uspto)**

---